

AfxParseURL

Does not fully comply with URL standards

Sean Barnum, Cigital, Inc. [vita¹]

Copyright © 2005 Cigital, Inc.

2005-10-03

Part "Original Cigital Coding Rule in XML"

Mime-type: text/xml, size: 5300 bytes

Attack Categories	<ul style="list-style-type: none">• Identity Spoofing• Path spoofing or confusion problem• Malicious Input• Resource Injection
Vulnerability Categories	<ul style="list-style-type: none">• Indeterminate File/Path• URL/Command Parsing• Privilege escalation problem• Unconditional
Software Context	<ul style="list-style-type: none">• String Parsing• Internet
Location	<ul style="list-style-type: none">• Afxinet.h
Description	<p>Use of AfxParseURL() is problematic because this function does not fully comply with standards.</p> <p>The AfxParseURL() method does an incomplete job of parsing URLs as specified by [RFC 1738 et al.] In particular, it does not understand the userid:password parameter and it does not handle special characters that might be used in injection attacks. Thus, a URL such as <code>http://joeblow:foobar@www.yoursite.com/../../../../foo+bar</code> will not be parsed properly. The <code>"joeblow:foobar@www.yoursite.com"</code> will not be parsed properly, and the <code>"+"</code> will not be translated into the URL-encoded form <code>"%2B"</code>.</p> <p>A program that tries to parse URLs this way will fail on complex, but legal, URLs.</p>

1. <http://buildsecurityin.us-cert.gov/bsi-rules/35-BSI.html> (Barnum, Sean)

	This could lead to a variety of problems. The program may access resources that it otherwise would not open because it fails to properly screen the URLs. Binary data passed in a URL would not be safely encoded either, meaning that it could be passed unmodified into the program.		
APIs	FunctionName	Comments	
	AfxParseURL	parses URL	
Method of Attack	Find a program that is vulnerable and enter binary data or other semantic attack data into a dialog that is parsed by this method.		
Exception Criteria			
Solutions	Solution Applicability	Solution Description	Solution Efficacy
	Whenever one is tempted to use AfxParseURL	Use AfxParseURL and specify RCU_BROWSER_MODE as your parser.	Relatively Effective.
Signature Details	BOOL AFXAPI AfxParseURL(LPCTSTR pstrURL, DWORD& dwServiceType, CString& strServer, CString& strObject, INTERNET_PORT& nPort);		
Examples of Incorrect Code	<pre>#include <afxinet.h> const char demoURL="http:// server.example.com/ foo/bar.html"; BOOL bDidItParse; DWORD dwServiceType; CString strServer; CString strObject; INTERNET_PORT nPort; bDidItParse = AfxParseURL(demoURL, &dwServiceType, &strServer, &strObject, &nPort);</pre>		
Examples of Corrected Code	<pre>#include <afxinet.h></pre>		

	<pre> const char demoURL="http:// server.example.com/ foo/bar.html"; BOOL bDidItParse; DWORD dwServiceType; CString strServer; CString strObject; INTERNET_PORT nPort; CString strUserid; CString strPasswd; // This is the critical difference here: flags tell it how to convert weird characters. DWORD dwFlags = ICU_BROWSER_MODE; bDidItParse = AfxParseURLEx(demoURL, &dwServiceType, &strServer, &strObject, &nPort, &strUserid, &strPasswd, dwFlags); </pre>					
Source Reference	<ul style="list-style-type: none"> • http://msdn.microsoft.com/library/default.asp?url=/library/en-us/vcmfc98/html/_mfc_afxparseurl.asp² 					
Recommended Resources	<ul style="list-style-type: none"> • MSDN web page about AfxParseURL³ • RFC 1738 specifying valid URL syntax⁴ 					
Discriminant Set	<table border="1"> <tr> <td data-bbox="590 1332 798 1413"> Operating System </td> <td data-bbox="798 1332 1000 1413"> <ul style="list-style-type: none"> • Windows </td> </tr> <tr> <td data-bbox="590 1413 798 1503"> Languages </td> <td data-bbox="798 1413 1000 1503"> <ul style="list-style-type: none"> • C • C++ </td> </tr> </table>	Operating System	<ul style="list-style-type: none"> • Windows 	Languages	<ul style="list-style-type: none"> • C • C++ 	
Operating System	<ul style="list-style-type: none"> • Windows 					
Languages	<ul style="list-style-type: none"> • C • C++ 					

Cigital, Inc. Copyright

Copyright © Cigital, Inc. 2005-2007. Cigital retains copyrights to this material.

Permission to reproduce this document and to prepare derivative works from this document for internal use is granted, provided the copyright and “No Warranty” statements are included with all reproductions and derivative works.

For information regarding external or commercial use of copyrighted materials owned by Cigital, including information about “Fair Use,” contact Cigital at copyright@cigital.com¹.

1. <mailto:copyright@cigital.com>

The Build Security In (BSI) portal is sponsored by the U.S. Department of Homeland Security (DHS), National Cyber Security Division. The Software Engineering Institute (SEI) develops and operates BSI. DHS funding supports the publishing of all site content.